Aerodynamic Shape Optimization Based on Free-form Deformation

Jamshid A. Samareh[*]
Multidisciplinary Optimization Branch
NASA Langley Research Center
Hampton, VA 23681

This paper presents a free-form deformation technique suitable for aerodynamic shape optimization. Because the proposed technique is independent of grid topology, we can treat structured and unstructured computational fluid dynamics grids in the same manner. The proposed technique is an alternative shape parameterization technique to a trivariate volume technique. It retains the flexibility and freedom of trivariate volumes for CFD shape optimization, but it uses a bivariate surface representation. This reduces the number of design variables by an order of magnitude, and it provides much better control for surface shape changes. The proposed technique is simple, compact, and efficient. The analytical sensitivity derivatives are independent of the design variables and are easily computed for use in a gradient-based optimization. The paper includes the complete formulation and aerodynamics shape optimization results.

Nomenclature

B	=	B-spline basis functions
C	=	product of B-spline basis functions
I	=	maximum number of points in u-direction
J	=	maximum number of points in v-direction
M	=	number of surface points
V	=	control points
W	=	weights
b	=	baseline grid
m,n	=	grid point number
r	=	coordinates (x, y, z)
v	=	design variable vector
ξ, η	=	surface bivariate coordinates

I. Introduction

Over the past several decades, aerodynamic shape optimization (ASO) has been successfully applied for two-dimensional and simple three-dimensional configurations. Four distinct steps are involved in ASO: 1) geometry parameterization; 2) surface grid generation; 3) volume grid generation, regeneration, or deformation; and 4) computational fluid dynamics (CFD) function and sensitivity analyses. Automatic grid generation and grid sensitivity analysis are two important issues for ASO. CFD grid generation for a full airplane model is time-consuming and costly. Detailed CFD grids based on a computer-aided design (CAD) model can take several weeks to develop and validate. Gradient-based optimization requires accurate sensitivity analysis for all four steps of ASO. Steps 2–4 can be automated for general configuration with unstructured grid technology and for specific configurations with structured grid technology. The level of effort for the geometric parameterization depends on the complexity of the vehicle configuration. This step is a labor-intensive process, and it is difficult to automate. For more details, readers are referred to an overview paper by this author, Ref. 1, on geometric modeling and grid generation for design and optimization. The purpose of this paper is to present a technique to simplify step 1 (shape parameterization) of ASO.

American Institute of Aeronautics and Astronautics

In general, shape parameterization techniques can be divided into eight categories[2]: basis vector, domain element, discrete, analytical, free-form deformation (FFD), partial differential equation (PDE), polynomial and spline, and computer-aided design (CAD). Among these categories, FFD and CAD appear to have the following desirable characteristics: 1) efficiency, 2) compactness, and 3) suitability for complex configurations. In addition, CAD-based parameterization techniques can achieve large geometry changes. This capability makes CAD-based techniques suitable for the conceptual design efforts. Another advantage of CAD-based optimization is the availability of a comprehensive set of geometric functionalities provided by commercial CAD systems. However, existing CAD systems lack tools for the automation of geometry abstraction and analytical sensitivity derivatives calculations.

Commercial CAD systems use the feature-based solid modeling (FBSM) concept to create parametric design models. To parameterize an existing complex model is still a challenging task with today's CAD systems, and the models created are not always good enough for automatic grid generation tools. Even though the use of parametric modeling in design would make FBSM tools ideal for optimization, existing FBSM tools are not capable of calculating sensitivity derivatives analytically. Computer codes for commercial CAD systems are very large; to differentiate an entire system with automatic differentiation tools may not be a trivial task. Therefore, calculation of analytical sensitivity derivatives of geometry with respect to design variables could prove to be difficult within a commercial CAD environment. For some limited cases, the analytical shape sensitivity derivatives can be calculated based on a CAD model; however, this method will not work under all circumstances. One difficulty is that, for perturbation of some parameters, the topology of the CAD part may be changed. Another way to calculate the sensitivity derivatives is to use finite differences, as long as the perturbed geometry has the same topology as the unperturbed geometry. Both methods—the analytical and finite-difference approximations—have their difficulties and limitations for CAD-based parameterization.

Free-form deformation is a subset of the soft object animation (SOA)[2] algorithms used in computer graphics for morphing images and deforming models. The FFD technique can achieve only small to medium geometry changes. Because the FFD parameterizes the grid, it eliminates the need for geometry abstraction and surface grid generation. Unlike the CAD-based technique, FFD does not require one to reverse engineer the original design parameters. The FFD technique is suitable for the preliminary design phase, where geometry changes are rather small. Another advantage of parameterizing the grid is that the grid topology stays fixed throughout optimization; hence, the grids can be regenerated (deformed) automatically. The disadvantage is that the large shape changes could produce unacceptable grids due to the fixed topology.

Our proposed technique uses nonuniform rational B-spline (NURBS) to model the geometry perturbations rather than the geometry itself. The parameterization of the shape perturbations is used in conjunction with the FFD. An appendix is included to provide a brief overview of NURBS representation and terminologies.

II. Shape Perturbations

During the preliminary design phase, numerical optimization starts with an existing design, and the goal is to redesign and improve performance by using numerical optimization. The geometric changes (perturbations) between initial and optimized shape are very small[2], but the difference in the performance can be substantial. An effective way to reduce the number of shape design variables is to parameterize the shape perturbations instead of parameterizing the shape itself. Throughout the optimization cycles, the analysis grid can then be updated as

$$r_n(v) = r_n^b + \Delta r_n(v) \tag{1}$$

where r_n is the design grid point n, v is design variable vector, r_n^b is the baseline grid, and $\Delta r_n(v)$ is the shape design perturbation for grid point n. Far fewer design variables are required to parameterize the shape perturbations, $\Delta r_n(v)$, than the baseline shape r_n^b itself. Because the parameterization is independent of grid topology, we can use this technique for structured and unstructured CFD.

III. Trivariate Volume-Based Deformation Techniques

The use of a trivariate volume-based deformation technique for shape optimization has been discussed in Ref. 3. The FFD algorithm described by Sederberg and Parry[4] is ideal for deforming the polygonal models (e.g., CFD grids). Like other SOA algorithms, this algorithm maintains the polygon connectivity, and the deformation is applied only to the vertices of the model. The FFD process is analogous to embedding the grid inside a block of clear, flexible plastic (deformation object) so that, as the plastic is deformed, the grid is deformed as well. Deformation of complex shapes may require several deformation objects. The shapes of these deformation objects are not arbitrary. In fact, the shapes are trivariate volumes and could range from a parallelepiped to a general

NURBS volume as shown in Fig. 1a. The grid is deformed by perturbing the vertices that control the shape of the deformation block (e.g., corners of the parallelepiped). Sederberg and Parry used a Bézier volume[4]. Coquillart at INRIA extended the Bézier parallelepiped to a nonparallelepiped cubic Bézier volume[5]. Lamousin and Waggenspack have further generalized this idea to NURBS volume[6](Fig. 1b), and they used multiple blocks to model complex shapes. This technique has been used for design and optimization by Yeh and Vance[7], Perry and Balling[8], and Ronzheimer[9]. For trivariate volume blocks, coordinates controlling the deformation are related through the mapping coordinates (ξ, η, ζ). These coordinates are used in both forward and backward mapping. For more details, readers are referred to the paper by this author, Ref. 2.

IV. Proposed Technique

The trivariate volume operates on the whole space, regardless of the representation of the deformed objects embedded in the space (Fig. 2). Aerodynamic shape optimization is concerned with modifying the aerodynamic surfaces, not the volumes. Because the trivariate representation operates on the whole space, the design variables may have no physical significance for the design engineers. This drawback makes it difficult to select an effective and compact set of design variables.

If we shrink the trivariate volume along one coordinate direction (e.g., ζ), it will be reduced to a bivariate FFD surface (Fig. 3). These surfaces can be laid (pasted) onto outer mold line (OML) surfaces. Then as these bivariate FFD surfaces move, the underlying grid points move as well. It is much more effective to modify the OML surfaces by these bivariate FFD surfaces than by trivariate volumes. The proposed technique retains the flexibility and freedom of trivariate volumes for CFD shape optimization, but it uses a bivariate representation. This reduces the number of design variables by an order of magnitude, and it provides a much better control for surface shape changes. Because the NURBS formulation is the most general free-form surface representation, it is used for the bivariate surfaces in this paper.

The implementation of the proposed algorithm is divided into the following steps:
1. Identify regions of OML that need to be designed or adjusted (e.g., fillet, inboard leading edge). These regions are identified by pasting a set of NURBS surfaces on the outer mold line (OML). We will refer to these surfaces as *marking surfaces*. These surfaces are used to *mark* and identify surface grid points that belong to individual design regions.
2. Determine the bivariate coordinates (ξ,η) of each grid point by projecting each one onto the marking surfaces.
3. Create design surfaces corresponding to each marking surface defined in step 1.

Step 1: Identify Design Regions

CAD systems are frequently used during the preliminary design phase to model and represent the OML geometry. These representations are either bivariate functions (e.g., NURBS, Bézier) or implicit functions (cylinders, spheres), which can be converted to bivariate functions. Figure 4 shows a CAD representation of a business jet model. The solid lines are the boundaries of CAD surfaces. Any bivariate surface can be used to represent the marking surface. We have used a NURBS representation, but the algorithm presented here is equally applicable for any bivariate surface representation. Figure 4 shows a marking surface for a business jet. Two important requirements are needed for creating marking surfaces: 1) they cannot overlap each other, but they can overlap CAD surfaces, and 2) they must be laid very close to OML surfaces within some tolerance. The number of control points and order of marking surfaces play no role in the parameterization process. However, the knot vector distribution could play a role in the quality of shape deformation.

Step 2: Project Grid Points on Marking Surfaces

Each baseline surface grid point is projected to all marking surfaces created in step 1. Then, each grid point is linked to the closest marking surface. The algorithm for grid point projection can be found in Ref. 10. After this step, each grid point in the design region is assigned a design surface (closest surface) and corresponding bivariate coordinates (ξ,η).

Step 3: Create NURBS Design Surfaces

The $\Delta r(v)$ in Eq. (1) is represented by a NURBS (see appendix for details on NURBS representation):

American Institute of Aeronautics and Astronautics

$$\Delta r_n(v) = \Delta r(v, \xi_n, \eta_n) = \sum_m^M C_m(\xi_n, \eta_n) v_m \qquad (2)$$

The coefficients C_m are products of B-spline basis functions. The only requirement for this design surface is that its knot vectors must span over bivariate coordinates (ξ, η) of all grid points belonging to individual design surfaces. We are free to pick the NURBS order and number of control points. The latter allows us to easily change the number design variable regardless of complexity of the baseline surfaces. As the control points of NURBS design surfaces move, the grid points belonging to the surface will move as well. The movement is based on an inverse mapping between the grid point bivariate coordinates and the NURBS surface. The bivariate coordinates of the grid points with respect to the marking surfaces are kept fixed through the optimization cycles. The movement may be unconstrained, or it can be constrained to move normally to the OML. The latter will reduce the number of control points (design variables) by a factor of three. All NURBS control points are initially set to zero, so the first computation gives the baseline grid.

The analytical sensitivity derivatives of grid points with respect to the design variables are determined from Eq.(2).

$$\frac{\partial r_n(v)}{\partial v_m} = \frac{\partial (\Delta r_n(v))}{\partial v_m} = C_m(\xi_n, \eta_n) \qquad (3)$$

The grid point sensitivity is independent of the design variables. Thus, we needed to calculate the sensitivity only once, at the beginning of the optimization. Then grid points can be updated as

$$r_n(v) = r_n^b + \sum_m^M C_m(\xi_n, \eta_n) v_m \qquad (4)$$

The C_m coefficients are precomputed, and they are used for sensitivity analysis as in Eq. (3) and for updating the grid point locations, as Eq. (4).

Results

Results are presented for a generic business jet and a morphing aircraft vehicle. Fig. 5 shows the deformed fillet for the business jet shown in Fig. 4. The marking surface was represented by a NURBS with 24 control points. The control points on the boundary were not allowed to move. The deformation was based on Eq.(4).

The second example is for the aerodynamic shape optimization of a morphing aircraft vehicle.[11] The overall vehicle optimization requires multiple disciplines (e.g., structures, control) and constraints to provide useful results for design. The purpose of these optimization examples is only to demonstrate the shape parameterization technique presented in this paper. Half of the vehicle was used for the aerodynamic optimization (Fig. 6). The FUN3D[12] CFD program was used for sensitivity analysis. This program is an unstructured Euler and Reynolds-averaged Navier-Stokes solvers, which computes the sensitivity derivatives efficiently and accurately. FUN3D is a node-based, implicit, upwind, finite volume solver on mixed-element unstructured grids across the speed range. The discrete adjoint sensitivity derivatives are hand coded. The Euler mesh was generated for a Mach 0.8 and angle of attack of 2.0. The mesh had 238362 nodes (1401823 tetrahedral cells). Fig. 7 shows the surface grid. Fig. 8 shows the marking NURBS surfaces used for the shape parameterization. There are 150 control points (design variables) available for optimization, but 54 design variables were active for shape optimization. Fig. 9 shows the surface grid points that are allowed to move. Two optimization calculations were performed on a 32-CPU Pentium IV cluster. The first optimization run was a lift-constrained drag minimization, resulting in a 23 percent reduction in drag (Fig. 10) while maintaining constant lift. The second case is an unconstrained L/D maximization, resulting in a 32 percent increase in L/D (Fig. 11). It took less than a day to prepare the CAD model, generate the Euler grid, set up the parameterization model, and set up CFD runs and optimization. It took another day to complete the calculations.

Conclusion

This paper presented a free-form deformation technique suitable for aerodynamic shape optimization. The present technique reduced the number of design variables by an order of magnitude. The analytical sensitivity derivatives were independent of the design variables and were easily computed for use in a gradient-based optimization. The aerodynamic shape optimization results demonstrated the effectiveness of the present formulation.

4

Appendix

This section contains a brief overview of the NURBS, and readers should consult Ref. 13 and 14 for a detailed discussion. Before I describe a NURBS surface representation, I will present a brief description of NURBS curves. A NURBS curve, $r(\xi)$, can be represented as

$$r(\xi) = \frac{\sum_{i=1}^{I} B_{i,p}(\xi) W_i V_i}{\sum_{i=1}^{I} B_{i,p}(\xi) W_i} \qquad (A.1)$$

The univariate coordinate, ξ, is bounded by $\xi_{min} \le \xi \le \xi_{max}$. The V_i are the control points (forming a control polygon), and W_i are the weights. The $B_{i,p}$ are the p-th degree B-spline basis functions defined on the nonperiodic and nonuniform knot vector (ξ)

$$\xi \in [\xi_{min} = \xi_1 \le \xi_2 \le \cdots \le \xi_k = \xi_{max}] \qquad (A.2)$$

where k is the number of knots. This completes the mapping between the one-dimensional univariate coordinate, ξ, and the three-dimensional Euclidean space, r. A NURBS curve has five important properties:

- o   It is invariant under linear transformation.
- o   A NURBS curve of order p, having no multiple interior knots, is p - 2 differentiable.
- o   The approximation is local in nature.
- o   A NURBS curve is contained in the convex hull of its control points.
- o   The NURBS approximation is variation diminishing.

To evaluate, the three-dimensional curve NURBS is commonly represented in homogeneous form as

$$r(x, y, x) \Leftrightarrow r^w(wx, wy, wz, w) \qquad (A.3)$$

So, the NURBS curve can conveniently be defined as a perspective map of its nonrational counterpart in four-dimensional space as

$$r^w = \sum_{i=1}^{I} B_{i,p}(\xi) V_i^w \qquad (A.4)$$

where V_i^w is defined as $V_i^w = \{x_i w_i, y_i w_i, z_i w_i, w_i\}$. The basis functions can be efficiently computed using DeBoor's algorithm[13].

A NURBS surface is a parametric surface and is defined as a bivariate function as

$$r(\xi, \eta) = \frac{\sum_{i}^{I} \sum_{j}^{J} B_{i,p}(\xi) B_{j,q}(\eta) W_{ij} V_{ij}}{\sum_{i}^{I} \sum_{j}^{J} B_{i,p}(\xi) B_{j,q}(\eta) W_{ij}} \qquad (A.5)$$

where the V_{ij} are the locations of NURBS control points, the W_{ij} coefficients are the weights, ξ and η are the bivariate coordinates, I and J are the numbers of control points in ξ and η directions, and $B_{i,p}(\xi)$ and $B_{j,q}(\eta)$ are the B-spline basis functions of degrees p and q respectively.
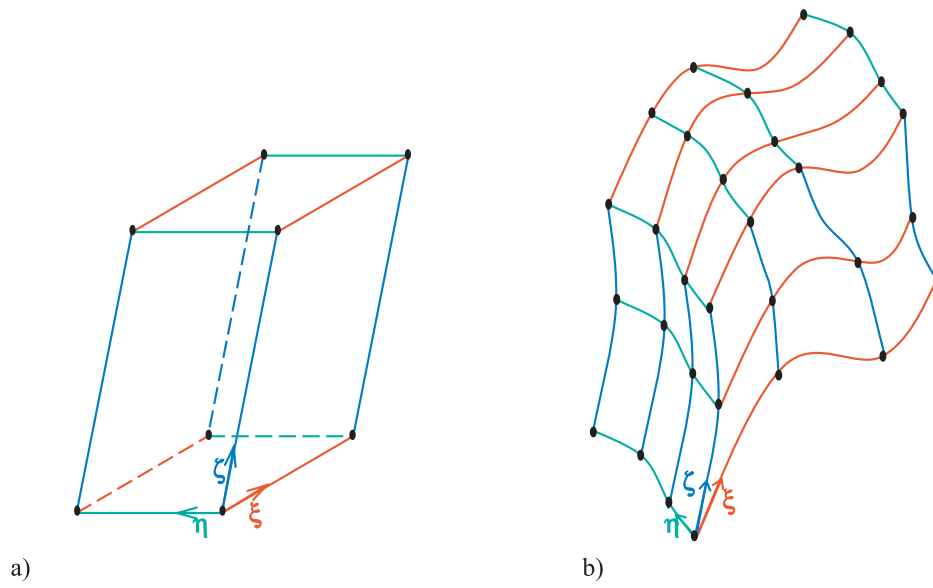
Similar to a NURBS curve, the NURBS surface can conveniently be defined as a perspective map of its nonrational counterpart in four-dimensional space as

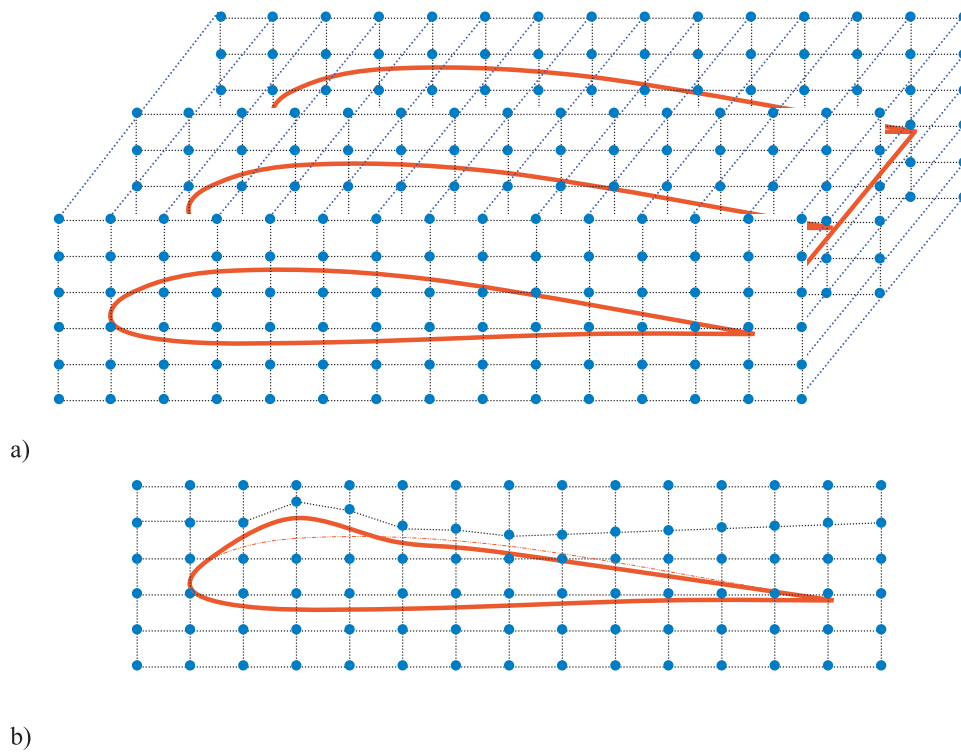$$r^w(\xi, \eta) = \sum_{i}^{I} \sum_{j}^{J} B_{i,p}(\xi) B_{j,q}(\eta) V_{i,j}^w \qquad (A.6)$$

This equation can be written in a compact form by collapsing the two summations into a single summation as

$$r(\xi, \eta) = \sum_{m}^{M} C_m(\xi, \eta) V_m^w, \quad \text{where} \quad m = i + I * (j-1), \quad M = I * J, \text{and}$$
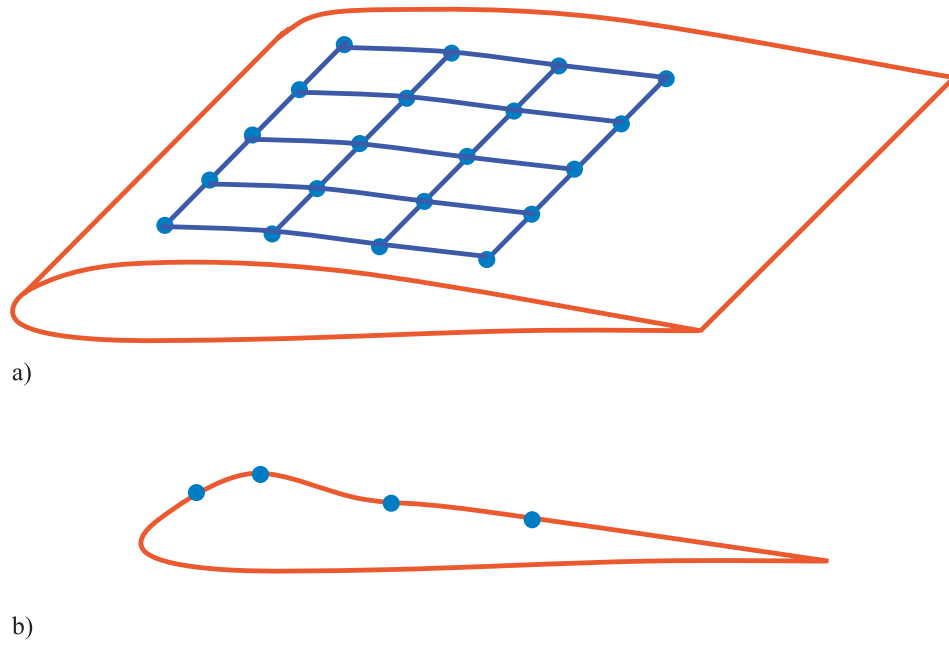
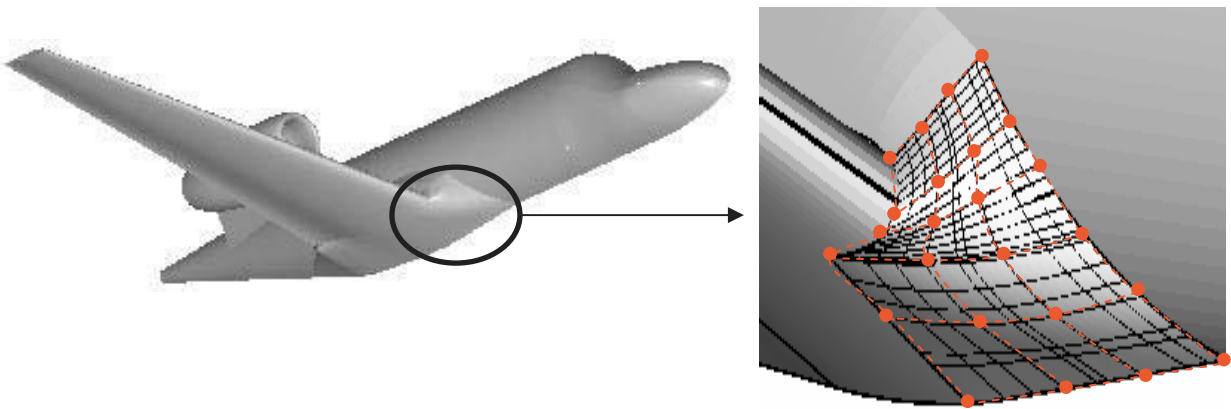$$C_m(\xi, \eta) = C_{ij}(\xi, \eta) = B_{i,p}(\xi) B_{j,q}(\eta) \qquad (A.7)$$

5

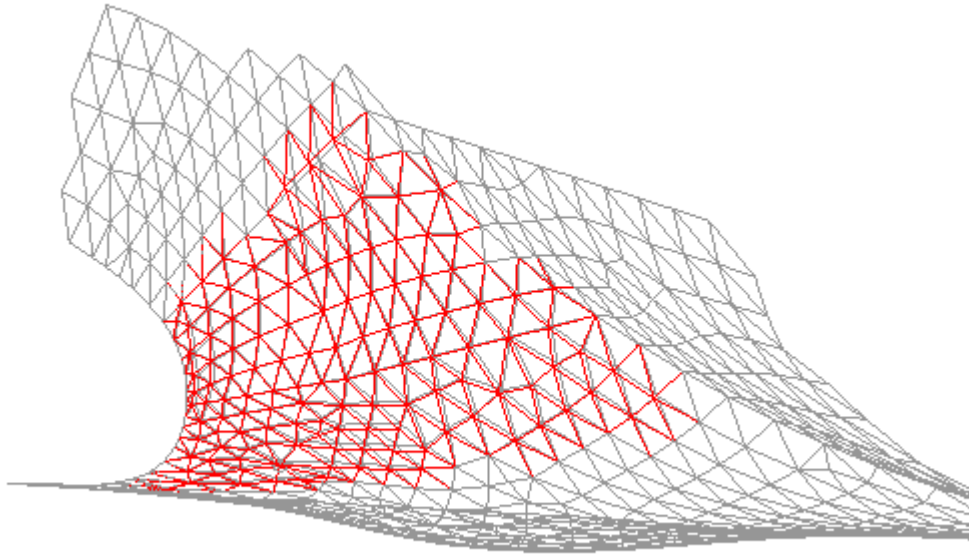**Figure 1: a) Parallelepiped volume, b) NURBS volume.**

a)

b)

Figure 2: a) Trivariate volume deformation, b) Deformed shape.

American Institute of Aeronautics and Astronautics
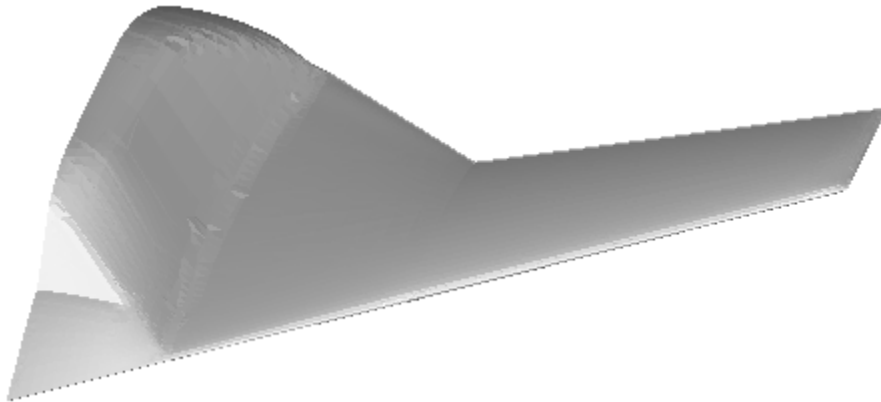
a)

b)

Figure 3: a) Bivariate surface deformation, b) Deformed cross section.

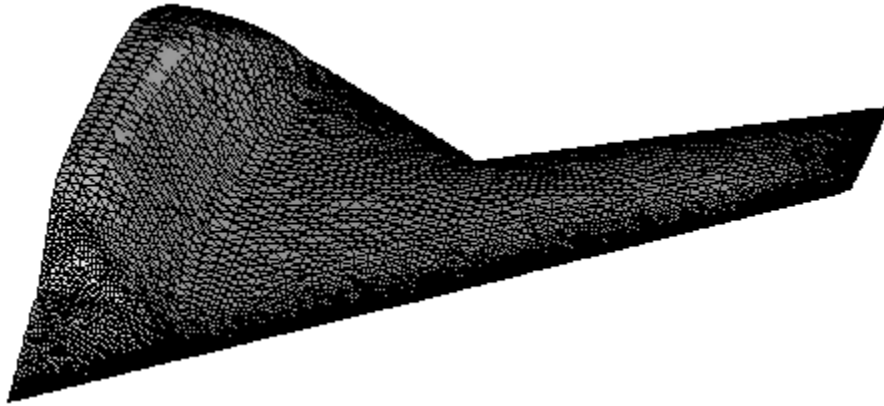**Figure 4: Parameterizing wing-fuselage fillet of a bushiness jet.**

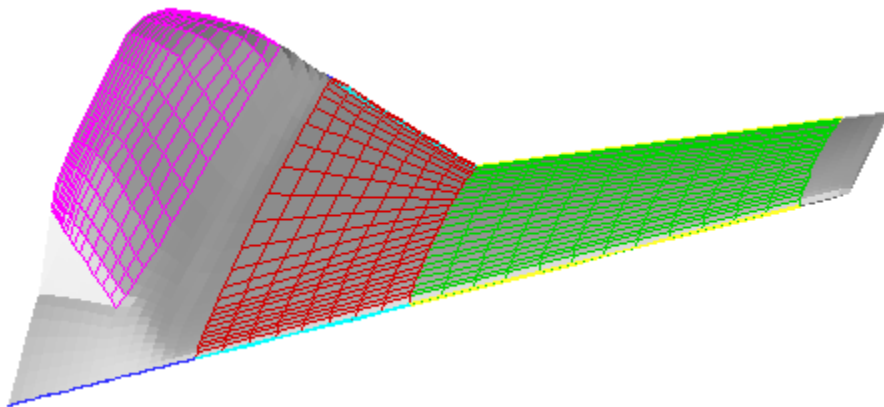**Figure 5: Free-form deformation of wing-fuselage fillet.**
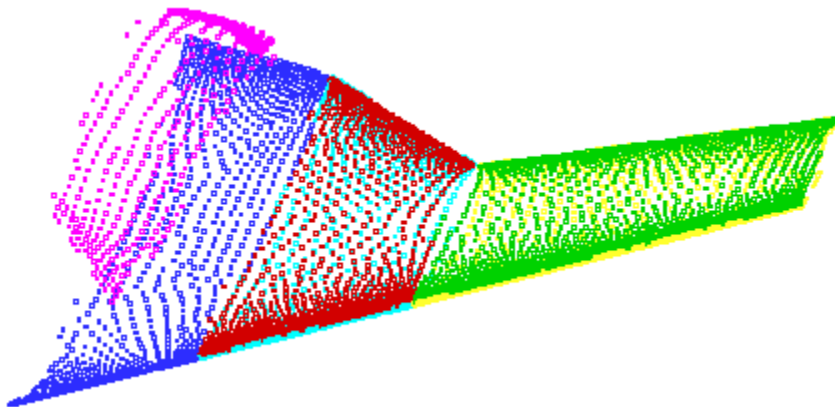

Figure 6: Morphing vehicle.

American Institute of Aeronautics and Astronautics

Figure 7: Surface grid.

Figure 8: NURBS marking surfaces.

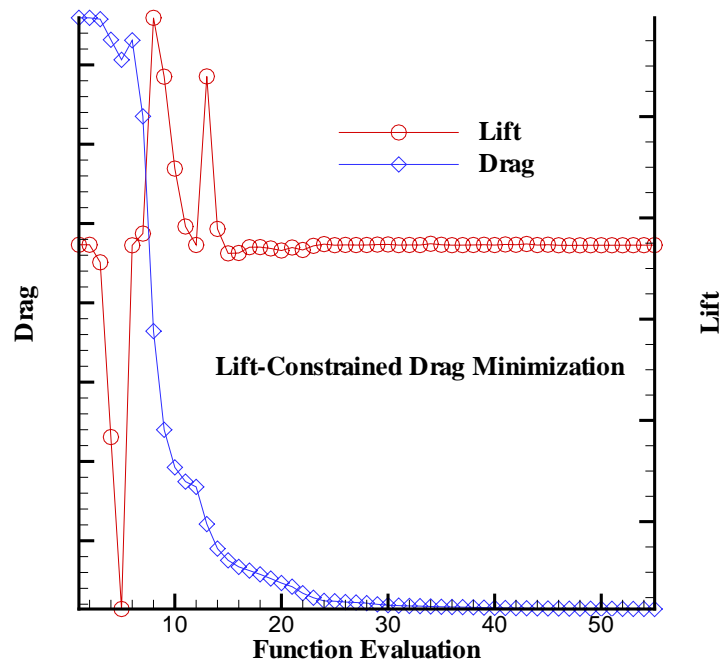**Figure 9: Surface grid points identified for deformation.**

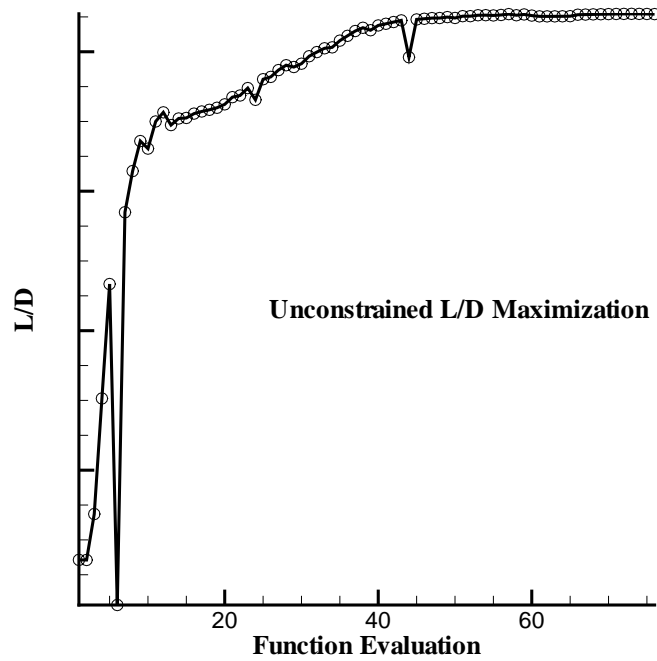**Figure 10: Lift-constrained drag minimization.**

Figure 11: Unconstrained L/D maximization.

American Institute of Aeronautics and Astronautics

Acknowledgments

American Institute of Aeronautics and Astronautics

References

[1] Samareh, J. A.: "Status and Future of Geometry Modeling and Grid Generation for Design and Optimization," J. Aircraft, Vol. 36, No. 1, Jan.-Feb. 1999, pp.97-104.

[2] Samareh, J. A., "Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization," *AIAA Journal*, May 2001, pp. 877–884.

[3] Samareh, J. A.: "Geometry Modeling and Grid Generation for Design and Optimization," ICASE/LaRC/NSF/ARO Workshop on Computational Aerosciences in the 21st Century, Hampton, VA, Apr. 22-24, 1998 (also in Computational Aerosciences in the 21st Century, Editors M. D. Salas and W. K. Anderson, Volume 8, November 2000, Kluwer Academic Publishers.

[4] Sederberg, T. W., and Parry, S. R., "Free-Form Deformation of Solid Geometric Models," *Computer Graphics*, Vol. 20, No. 4, 1986, pp. 151–160.

[5] Coquillart, S., "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," *SIGGRAPH*, Vol. 24, No. 4, 1990, pp. 187–196.

[6] Lamousin, H. J., and Waggenspack, W. N., "NURBS-Based Free-Form Deformation," *IEEE Computer Graphics and Applications*, Vol. 14, No. 6, 1994, pp. 95–108.

[7] Yeh, T. P., and Vance, J. M., "Applying Virtual Reality Techniques to Sensitivity-Based Structural Shape Design," *Proceedings of 1997 ASME Design Engineering Technical Conferences*, No. DAC-3765 in DETC97, Sept. 1997, pp. 1–9.

[8] Perry, E., and Balling, R., "A New Morphing Method for Shape Optimization," AIAA-98-2896, June 1998.

[9] Ronzheimer, A., "Post-Parameterization of Complex CAD-Based Aircraft-Shapes Using Freeform Deformation," *8th International Conference on Numerical Grid Generation in Computational Field Simulations*, June 2–6, 2002, Honolulu, Hawaii, USA.

[10] Samareh, J., "Triangulation of NURBS Surfaces," *4th International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Swansea, Wales, UK, April 1994, pp. 377–388. URL: http://techreports.larc.nasa.gov/ltrs/PDF/conf-nurbs.pdf

[11] Love, M. H., Zink, P. S., Stroud, R. L., Bye, D. R., and Chase, C., "Impact of Actuation Concepts on Morphing Aircraft Structures," AIAA-2004-1724, April 2004.

[12] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155–1163.

[13] Farin, G., Curves and Surfaces for Computer Aided Geometric Design, Academic Press, New York, 1990.

[14] Piegl, L. and Tiller, W., The NURBS Book, Springer-Verlag, 1997.